SHEET – POINTERS PART I

Q1 -Write a C function to add two numbers using pass by reference.

Test Data :

Input the first number : 5

Input the second number : 6

Expected Output :

The sum of the entered numbers is : 11

Q2 - Write a C function to find the maximum number between two numbers using pass by reference.

Test Data :

Input the first number : 5

Input the second number : 6

Expected Output :

6 is the maximum number.

Q3 -Write 2 Functions in C , one to store n elements in an array and other one print the elements.
NOTE : You should store the elements in the array and print them using pointer to array

Functions prototype :

```
void store_in_array(int *arr,int n);

void print_array(int *arr,int n);
```

Test Data(store_in_array function) :

```
Input 5 number of elements in the array :

element [0] : 5

element [1] : 7

element [2] : 2

element [3] : 9

element [4] : 8
```

Expected Output (print_array function):

```
 The elements you entered are :

element [0] : 5

element [1] : 7

element [2] : 2

element [3] : 9

element [4] : 8
```

Q4 Write a program in C to compute the sum of all elements in an array using pointers.

Test Data :

Input the number of elements to store in the array (max 10) : 5

Input 5 number of elements in the array :

element - 1 : 2

element - 2 : 3

element - 3 : 4

element - 4 : 5

element - 5 : 6

Expected Output :

The sum of array is : 20

BOUNCE QUESTION: -

write a c function that takes two numbers, and returns their sum, their difference, their product.

MCQ QUESTIONS: -

Q1 : -

```
#include <stdio.h>
void foo(int *);
int main()
{
    int i = 10, *p = &i;
    foo(&i);
}
void foo(int *p)
{
     printf("%d\n", *p);
 }
```

a) 0
b) 1
c) 10
d) -1

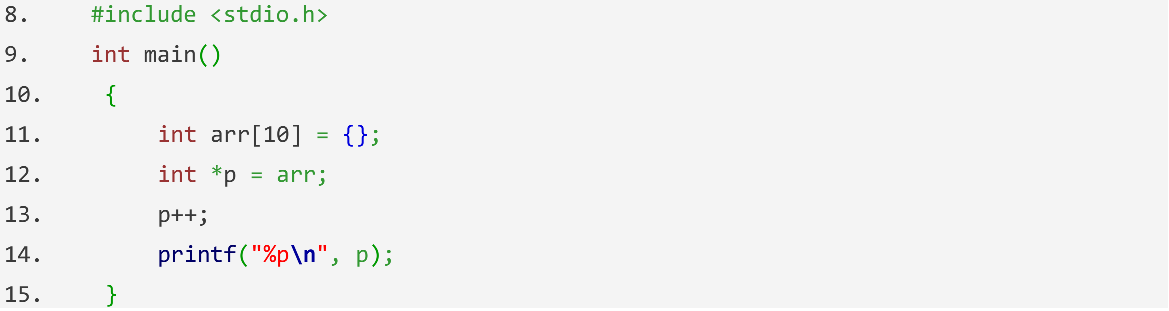
Q2 : -

```
#include <stdio.h>
int main()
{
    int i = 0x12345678;
    char *p = &i;
    printf("%x\n", *p);
}
```

Note : Our toolchain is BIG ENDIAN TOOLCHAIN

a) Unspecified value
b) 12
c) 78
d) 12345678

Q3 : -



```
#include <stdio.h>
int main()
 {
     int arr[10] = {};
     int *p = arr;
     p++;
     printf("%p\n", p);
 }
```

NOTE : The base address of array(address of arr[0] is 0x1234)

a)0x1235
b)0x1233
c)0x1236

d)0x1238

SEARCH TASK : -

A) What is the difference between ptr++ , *ptr++ and (*ptr)++? Show an example.

B) can we do any arithmetic operation on the array name? example :
int arr[10];

Arr++;

Is that operation valid?

GOOD LUCK